Design Document for The Flying Dutchman

Max Block   Linda Eriksson   Daniel Llatas Spiers
Joakim Jrvinen   Christian Trnqvist   Joel Wiklund

Group Stormrider
Version 1.0

February 2015

Abstract

Abstract.

Contents

Changelog

Version 1.0: Created document.

Part I
Design Overview

1 Personas

1.1 Bartender

Caesar is a 21 years old man, who just finished bartending school. This is the first time ever that he is working at The Flying Dutchman. Although he just finished bartending school, he is also doing academic studies so his work at The Flying Dutchman is not full time. His intention is to work at nights and study daytime in order to finance his studies.

1.2 Professor

Bengt is a 53 year old man from Sweden. He is a professor in the field of computer science. He is a very social person who spend a lot of his spare time at The Flying Dutchman where he is on the list of so called VIPs, who have some special privileges over other customers. Bengt enjoys drinking beer and especially trying out rare special beers which may not be available in your local liquor store. Socially, he enjoy talking with students at the bar about their life choices and giving them some valuable advice.

1.3 Student

Alice is a 23 year old girl who is an international student from USA that loves to be around people. She enjoys going out partying and wants to do it as often as possible. However, as an international student, she has a very limited budget and thus have to think economically when going out. She really likes being in Sweden and plans on learning the language so that she can settle down there.

1.4 Manager

Marie is a 35 year old Swedish woman with great experience as a pub manager. During her twenties, she was a student at Uppsala University, however, studying was not the thing that got the most of her attention but rather spending time and working at nations. She worked so much that she had to take a break from her studies. She never got back from her break but she had figured that she wanted to work full time in bars. This resulted in her applying to The Flying Dutchman where she got employed and it did not take long before she became the manager.

2 Scenarios

2.1 Student

- Goes to the bar and asks the bartender for the cheapest beer possible. The bartender, who has just started his career, doesnt know the menu and has to search in the system for the cheapest beer. He finds the beer, bring it and gives it to Alice. Alice pays both by cash and by card.

- Alice now wants a second round and orders two beers. This time she tries to pay only by card. But she doesnt have enough money, so the bank rejects the payment. She tells the bartender to remove one beer and tries to pay again. Once again the payment doesnt go through so the bartender cancels the order.

2.2 Professor

- He goes to the VIP terminal, logs in and looks at the available beers. So he picks one of his favorite beers, a nice Belgian Ale. He checks that he has enough credit and then checkouts. After that, he goes and grabs the beer.

- The professor wants to become a VIP customer. So he goes to the manager and asks to become a VIP. The manager approves and gives Bengt a register code. Bengt goes to the VIP terminal, type in the code and fills in a form with his personal data.

- The professor has used all his credit. He goes to the bartender to refill his credit.

2.3 Bartender

- A customer comes and wants to know which beers from "Brewdog" are available at the bar. So Caesar searches for "Brewdog", and the system displays a list of these beers. The customer says what he wants, the bartender selects it and goes to checkout.

- A customer wants a Heineken. Caesar selects Heineken as it is the most common beer and goes to checkout.

2.4 Manager

- The system makes a suggestion for the beers to ordered. And then the manager can edit by adding/subtracting numbers, add a new type of or deleting items. He then approves the order.

- The manager can add a new bartender.

- The manager can add a new regular.

3   Requirements specification

4   Database API

4.1   admin

4.1.1   inventory_get

Gives a list of all drinks available in the system.
Parameters:
none
Returns:
JSON-object:
- type: "inventory_get"
- payload[]
    - namn
    - namn2
    - sbl_price
    - pub_price
    - beer_id
    - count
    - price
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&
action=inventory_get
```

4.1.2   purchases_get_all

Gives a list of all purchases made by all users.
Parameters:
none
Returns:
JSON-object:
- type: "purchases_get"
- payload[]
    - namn
    - namn2
    - transaction_id
    - user_id
    - beer_id
    - timestamp
    - price
    - first_name
    - last_name
    - username

Usage:
`http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&`
`action=purchases_get_all`

4.1.3   payments_get_all

Returns a list of payments made by all users.
Parameters:
none
Returns:
JSON-object:
- type: "payments_get_all"
- payload[]
  - admin_username
  - timestamp
  - amount
  - admin_id
  - username
  - first_name
  - last_name
Usage:
`http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&`
`action=payments_get_all`

4.1.4   iou_get_all

Returns all users and amounts they have at their disposal. For users with a negative balance it means that they have a debt.
Parameters:
none
Returns:
JSON-object:
- type: "iou_get_all"
- payload[]
  - user_id
  - first_name
  - last_name
  - assets
Usage:
`http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&`
`action=iou_get_all`

4.1.5 user_edit

Updates user information. All user information is required as additional parameters.
Parameters:
new_username
new_password
first_name
last_name
email
phone
Returns:
JSON-object:
- type: "User user added"
- payload[]
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&
action=user_edit&new_username=newusername&new_password=newpassword&
first_name=firstname&last_name=lastname&email=email&phone=phone
```

4.1.6 inventory_append

Updates the number of bottles available in stock and the price for a particular beer. The id of the beer to be updated is required along with its price and current amount of available bottles.
Parameters:
beer_id
amount
price
Returns:
JSON-object:
- type: "empty"
- payload[]
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=ervtod&password=ervtod&
action=inventory_append&amount=amount&price=price
```

4.2 user

4.2.1 purchases_get

Gives a list of all purchases made by the specified user.
Parameters:
none
Returns:

JSON-object:
- type: "purchases_get"
- payload[]
    - namn
    - namn2
    - transaction_id
    - user_id
    - beer_id
    - timestamp
    - price

Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
action=purchases_get
```

4.2.2   purchases_append

Adds a purchase of one beer to the system for the specified user. The id of the
beer purchased beer is a required additional parameter.
Parameters:
beer_id
Returns:
JSON-object:
- type: empty
- payload[]
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
action=purchases_append&beer_id=beerid
```

4.2.3   payments_get

Returns a list of payments made by the specified user.
Parameters:
none
Returns:
JSON-object:
- type: "payments_get_all"
- payload[]
    - transaction_id
    - user_id
    - admin_id
    - amount
    - timestamp

Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
```

```
action=payments_get
```

4.2.4 payments_append

Adds a payment of specified amount to the system for the specified user. The amount is a required additional parameter.
Parameters:
amount
Returns:
JSON-object:
N/A
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
action=payments_append&amount=amount
```

4.2.5 iou_get

Returns the total amount that the specified user has at his disposal for buying beer. A negative balance means that the user has a debt.
Parameters:
none
Returns:
JSON-object:
- type: "iou_get"
- payload[]
    - user_id
    - first_name
    - last_name
    - assets
Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
action=iou_get
```

4.2.6 beer_data_get

Returns all information available in the system for a specified beer. This includes name, price, alcohol volume, etc. The id of the beer for inquiry is a required additional parameter.
Parameters:
beer_id
Returns:
JSON-object:
- type: "beer_data_get"

- payload[]
    - nr
    - artikelid
    - varnummer
    - namn
    - namn2
    - prisinklmoms
    - volymiml
    - prisperliter
    - saljstart
    - slutlev
    - varugrupp
    - forpackning
    - forslutning
    - ursprung
    - ursprungslandnamn
    - producent
    - leverantor
    - argang
    - provadargang
    - alkoholhalt
    - modul
    - sortiment
    - ekologisk
    - koscher

Usage:
```
http://pub.jamaica-inn.net/fpdb/api.php?username=aamsta&password=aamsta&
action=beer_data_get&beer_id=beerid
```

Part II
Implementation